อาจหาญ สัตยารักษ์

SOFT MAIL

ป้ายประกาศ(2)
: การแสดงอักขระบนจอภาพ

โปรแกรมป้ายประกาศ สำหรับภาษาไทย ที่สร้างตัวอักขระไทย ได้ตามแต่ต้องการ

กว่าจะมาพบกันวันนี้คิดว่าผู้อ่านคงได้สร้างอักขระกันไปคนละชุดแล้ว และคงอยากจะทราบแล้วว่าฟอนต์ที่เราสร้างเก็บไว้นี้ เราจะนำมาแสดงตัวอักขระได้สักกี่แบบกัน

การกำหนดตัวแปรใช้งาน

เพื่อที่เราจะสามารถเรียกใช้ฟอนต์ของอักขระได้หลายฟอนต์ จึงจำเป็นต้องกำหนดชนิดของข้อมูลในอักขระใหม่โดยเปลี่ยนมาเป็นการใช้ตัวแปรไดนามิกที่อาศัยในหน่วยความจำที่เรียกว่า ฮีพ (heap) มาช่วยงานแทน ซึ่งได้เคยกำหนดไว้แล้วในตอนที่ 1 ของไมโครเล่ม 74 ดังนั้นจึงขอเสนอการกำหนดตัวแปรใหม่ทั้งหมดดังรูปที่ 1 ซึ่งผลของการเปลี่ยนชนิดของตัวแปรนี้จึงทำให้ต้องเปลี่ยนจากชุดคำสั่ง loadfile: มาเป็นของคำสั่งใหม่ตามรูปที่ 1 ด้วย นอกจากนี้ยังมีชุดคำสั่งที่ไม่ได้แสดงอีกคือ opengraph; readchr; clsGR;, setpattern; เพราะได้เขียนถึงมาแล้วในบทก่อน

นอกจากนี้ยังมีตัวแปรบางตัวได้กำหนดไว้ล่วงหน้าสำหรับตอนที่ 3 ด้วย ดังนั้นรูปที่ 1 นี้จะใช้ในตอนที่ 3 ด้วย (จึงจะไม่นำไปแสดงในบทความอีก) ตัวแปรที่เปลี่ยนมาใช้ข้อมูลชนิดพอยน์-เตอร์คือ ShapeHead และ ShapePoly และที่กำหนดใหม่คือ YScope ซึ่งจะนำไปใช้ในกรณีที่ต้องการให้แสดงอักขระในแนวโค้งเว้าหรือโค้งนูน ข้อมูลประเภทนี้จะต้องมีการจองหน่วยความจำฮีพเพื่อรับด้วยคำสั่ง mark และ new ด้วยเสมอ

ตัวแปรที่กำหนด

ตัวแปรที่กำหนดไว้มีที่สำคัญดังนี้

1. xStart นำไปกำหนดการแสดงข้อความแบบเฉียงขึ้นและเฉียงลง

2. Ypoint นำไปตรวจสอบความกว้างของตัวอักขระแต่ละตัวเพื่อที่จะได้แสดงอักขระตัวต่อไปในจุดที่เหมาะสม ทำให้ได้ข้อความที่ปรับระยะห่างตามตัวอักขระแต่ละตัวได้

```
PROGRAM Show_ShapeTable_THAI_BANNER;
uses crt,dos,graph;

type  PolygonType=array[1..255] of pointtype;
      PolyByte=array[1..2] of shortInt;
      PolyInt=array[1..2] of integer;
      Shape_Head=^ShapeH;     ShapeH=array[1..255] of Integer;
      Shape_Poly=^Shape;      Shape=array[1..10000] of PolyByte;
      Y_Scope=^Scope;         Scope=array[0..4000] of integer;
      Data_Banner=record
                    x,y:integer; str:string;
                    size,patt:byte;
                  end;

const patternMe:array[0..9] of FillpatternType=
  (($FE,$EF,$FE,$EF,$FE,$EF,$FE,$EF),($18,$3C,$7E,$FF,$FF,$7E,$3C,$18),
   ($77,$77,$77,$77,$77,$77,$77,$77),($95,$59,$95,$59,$95,$59,$95,$59),
   ($E7,$7E,$E7,$7E,$E7,$7E,$E7,$7E),($6D,$D6,$6D,$D6,$6D,$D6,$6D,$D6),
   ($B3,$3B,$B3,$3B,$B3,$3B,$B3,$3B),($5C,$FF,$C3,$C3,$C3,$C3,$FF,$5C),
   ($5C,$E7,$7E,$C5,$E7,$7E,$C5,$E7),($18,$3C,$7E,$FF,$7E,$5A,$18,$18));

var
    SizeOfShape,xStart,Ypoint,PageNo:word;
    ColorBox,ColorBg,PattNo,ShapeNo,TableNo,
    font,High,Enlarge,MaxFont,PlayMode,NoSTR:byte;
    PolyPoint:array[1..255] of PolyInt;
    ShapeHead:array[1..4] of Shape_head;
    ShapePoly:array[1..4] of Shape_Poly;
    Funckey,ch:char;          Monoscreen:boolean;   Regs:registers;
    XRight,Position:Integer;  Xpoint:array[1..10] of Integer;
    up,down,Italic,Shadow,Symbol:boolean;  Ywave:array[0..719] of integer
    YScope:array[1..10] of Y_Scope;
    BannerData:array[1..10] of Data_Banner;

PROCEDURE LoadFile;
var f:file; PointStart:word;  STT:string;
begin
 MaxFont:=2;
for font:=1 to Maxfont do
 begin
  mark(ShapeHead[font]); new(ShapeHead[font]);
  mark(ShapePoly[font]); new(ShapePoly[font]);
  str(font,STT);   assign(f,'banner'+STT+'.fon'); reset(f,1);
  blockread(f,ShapeNo,1);        blockread(f,PointStart,2);
  blockread(f,SizeOfShape,2);    blockread(f,ShapeHead[font]^,510);
  blockread(f,ShapePoly[font]^,SizeOfShape);
  close(f);
 end;  font:=1;
end;
```

รูปที่ 1 ตัวแปรสำหรับการใช้งานและชุดคำสั่ง loadfile

3. font ตัวแปรกำหนดแบบตัวอักขระที่จะนำมาแสดงบนจอภาพกำหนดไว้สูงสุด 4 ฟอนต์ กำหนดค่าสูงสุดของจำนวนฟอนต์ด้วยตัวแปร MaxFont ในโปรแกรมกำหนดไว้เพียง 2 ฟอนต์

4. High, Enlarge ตัวแปรกำหนดการแสดงตัวสูง 2 เท่าและตัวกว้าง 2 เท่า

5. PlayMode ตัวแปรกำหนดรูปแบบการแสดงข้อความที่มีค่า 0-5 ดังนี้

0 – แสดงข้อความตามแนวนอนตามปกติ

1 – แสดงข้อความแบบการเคลื่อนที่ของคลื่นโดยใช้ ^PW (#23) กำหนด

2 – แสดงข้อความแบบโค้งนูนขึ้นโดยใช้ ^PC (#3) กำหนด

3 – แสดงข้อความแบบโค้งเว้าลงโดยใช้ ^PV (#22) กำหนด

4 – แสดงข้อความแบบเฉียงลงโดยใช้ ^PD (#4) กำหนด

5 – แสดงข้อความแบบเฉียงขึ้นโดยใช้ ^PU (#21) กำหนด

6. Xpoint, YScope กำหนดสำหรับนำไปหาระยะที่เหมาะสมในการแสดงข้อความแบบโค้งนูนหรือโค้งเว้า

7. Ywave กำหนดไว้รองรับระยะที่คำนวณได้ของการแสดงข้อความแบบคลื่น

8. Italic, Shadow กำหนดสำหรับการแสดงข้อความแบบตัวเอนและตัวนูน

9. bannerData กำหนดไว้สำหรับเก็บรายละเอียดต่าง ๆ ของข้อความที่จะนำมาแสดง (จะใช้ในตอนที่ 3)

10. symbol กำหนดไว้รองรับการแสดงอักขระลำดับที่ 164 เป็นต้นไป ซึ่งควบคุมด้วยรหัส ^PP ชุดคำสั่ง

loadfile; มีส่วนเปลี่ยนแปลงที่ควรรู้ดังนี้

10.1 กำหนดชื่อไฟล์ของอักขระในชื่อ banner1.fon-banner4.fon จึงให้ก๊อบปี้ไฟล์มาใส่ในชื่อที่กำหนดนี้ (ในตัวอย่างกำหนดเรียกใช้เพียง 2 ฟอนต์ (MaxFont=2)) ถ้ามีฟอนต์มากกว่านี้ก็ให้กำหนดตามที่มีใช้

10.2 ทำการอ่านไฟล์ข้อมูลเข้ามาโดยต้องมีการกำหนดตำแหน่งของหน่วยความจำฮีพด้วยคำสั่ง mark, new ด้วย ซึ่งคือตัวแปร ShapeHead และ ShapePoly

10.3 กำหนดชื่อไฟล์ด้วยการนำเอาตัวเลข (ในรูปของอักขระ) มาต่อท้ายชื่อไฟล์ banner แล้วตามด้วยสกุล (.fon) เพื่อเปิดไฟล์ข้อมูลอ่านข้อมูลทั้งหมดเข้ามา

ขนาดของตัวแปร SizeofShape นั้นจะมีค่าเท่ากับ (ขนาดของไฟล์-515)/2 ซึ่งถ้าตรวจดูทุกฟอนต์ที่ใช้ก็สามารถนำไปกำหนดขนาดของมิติของตัวแปร Shape (ในบรรทัดที่ 8-9) ให้เหมาะสมได้

การคำนวณหาค่าตัวแปร

เพื่อให้การแสดงข้อความมีรูปแบบแตกต่างกัน เราจำเป็นต้องเขียนชุดคำสั่งเพื่อเก็บค่าบางอย่างไว้ใช้งานตามรูปแบบที่เรากำหนดไว้ ซึ่งตามตัวอย่างนี้เราจะต้องคำนวณหาค่าของจุดที่จะไปรองรับการแสดงผล 3 แบบ คือ การแสดงข้อความแบบการเคลื่อนที่ของคลื่น และการแสดงข้อความแบบโค้งนูนและโค้งเว้า

ก. การแสดงข้อความให้เป็นแบบคลื่น (sine wave) มีวิธีการคำนวณดังนี้

1. กำหนดให้แสดงรูปคลื่น 2 ลูกต่อหนึ่งหน้าจอภาพ (ค่าตามแกนนอนจาก 0-GetMaxX) นั่นคือต้องสร้างรูปคลื่นให้เปลี่ยนเฟสไป 720 องศา หรือ 4π เรเดียนในหนึ่งหน้าจอ

2. ระยะที่เราจะให้เปลี่ยนไปก็คือค่าในแกนตั้ง (แกน y) ดังนั้นจึงกำหนดตัวแปร Ywave มารองรับค่าที่คำนวณได้ โดยจะคำนวณหาค่า Ywave นี้ในทุก ๆ ค่าของ x บนแกนเพื่อนำไปรวมกับตำแหน่งต่าง ๆ ในอักขระแต่ละตัวตามแกนตั้ง

3. ถ้าวาดรูปตัวอักษรในตำแหน่งที่คำนวณได้ก็จะได้ข้อความแสดงแบบคลื่นตามต้องการ

การคำนวณหาค่า Ywave นี้เป็นไปตามชุดคำสั่ง wavepoint ในรูปที่ 2

ข. การแสดงข้อความแบบโค้งนูนและโค้งเว้ามีวิธีการดังนี้

1. นำข้อความมาคำนวณหาระยะตามแกน x โดยประมาณที่จุด (เพื่อทราบความโค้งที่เหมาะสมกับข้อความ) ค่านี้เป็นค่าประมาณเพราะตัวอักขระแต่ละตัวนั้นจะมีความกว้างไม่เท่ากัน การคำนวณนี้จะต้องคิดขนาดของตัวขยายที่กินความกว้างเพิ่มขึ้นสองเท่าด้วย

2. คำนวณหาค่าความโค้ง (ความสูงตามแกนยืน) โดยหารัศมีของวงรีครึ่งส่วนนำมาเก็บในตัวแปร YScope ที่กำหนดเป็นตัวแปรสองมิติ เพราะอาจมีข้อความมากกว่าหนึ่งข้อความซึ่งมีความยาวที่ไม่เท่ากัน

การคำนวณค่า YScope นี้อยู่ในชุดคำสั่ง ScopePoint ในรูปที่ 2 เช่นกัน

แสดงอักขระบนจอภาพ

การแสดงอักขระบนจอภาพให้ตรงกับข้อความที่เรากำหนดนั้น ผู้เขียนขอใช้ภาษาไทยรหัสเกษตรเป็นต้นแบบจากที่กำหนดอักขระที่สร้างไว้ในตอนที่ 1 ในตารางรูปที่ 7

ลองมาดูภาษาอังกฤษ เช่นอักษร A จะอยู่ลำดับที่ 33 นั่นคือ หาได้จาก

```
PROCEDURE ScopePoint(content:string; size:byte);
var k,XMax:word; stt:string;

      FUNCTION FindLength(Content:string):word;
        var k,count:word;  Ccount:byte;
        begin  count:=0;  Ccount:=1;
            for k:=1 to length(Content) do
              begin
                if content[k]=#5 then Inc(Ccount);
                if Ccount>2 then Ccount:=1;
                if content[k] in [#32..#214] then inc(count,Ccount);
              end;
            FindLength:=Count*Size*20;
        end;
begin
  XMax:=FindLength(Content);
  for k:=0 to XMax+50 do
    Yscope[NoSTR]^[k]:=round(50*sin(k/XMax*pi));
end;

PROCEDURE WavePoint;
var k:word;  ShapeSTR:string;
begin
  for k:=0 to getMaxX do
        Ywave[k]:=round(24*sin(k/getMaxX*4*pi));
end;
```

รูปที่ 2 ชุดคำสั่งคำนวณหาจุดต่าง ๆ เพื่อแสดงข้อความแบบโค้งและแบบคลื่น

รหัสแอสกี−32 จึงได้ลำดับของ A เป็น 65−32=33 เป็นต้น ตัวอักขระอื่น ๆ ก็มีลักษณะแบบเดียวกัน นั่นคือ ถ้ารหัสแอสกีนั้นเป็นภาษาอังกฤษก็หาลำดับในแฟ้มข้อมูล (โปรแกรมใช้ตัวแปรชื่อ TableNo) ด้วยการลบรหัสแอสกีออกด้วยเลข 32 สังเกตรหัสภาษาไทยพบว่า อักษร ก อยู่ที่ลำดับ 96 ดังนั้นภาษาไทยจึงหาตัวเลขที่เอามาลบออกได้จาก 161−96=65 (161 คือรหัสแอสกีของ ก)

หากประสงค์จะใช้อักขระลำดับที่ 164 เป็นต้นไปทำได้โดยใช้รหัสควบคุม ในที่นี้ผู้เขียนใช้ ^PP เข้า-ออกอักขระกลุ่มนี้โดยใช้คู่กับอักขระภาษาอังกฤษ ดังนั้นค่าของอันดับที่จึงต้องบวกเพิ่มจากรหัสแอสกีอีก 164−1=163 (อักขระ ! คือข้อมูลลำดับที่ 1 ของไฟล์ตัวอักษร) แต่ฟอนต์ที่ให้มาพร้อมซอฟต์เมล์นั้นยังไม่ได้ทำอักขระลำดับที่ 164-254 ให้ เพียงแต่ได้กำหนดส่วนของตัวแปร shapeHead[164]−ShapeHead[255] ให้เรียกอักขระภาษาอังกฤษมาแสดงแทน (ตัวแปร ShapeNo ยังคงเป็น 164) หากต้องการเขียนส่วนนี้เพิ่มก็ให้ใช้โปรแกรมตอนที่หนึ่งสร้างอักขระตามความต้องการของท่านเอง

เมื่อเราสามารถหาลำดับที่ของตัวอักขระได้ เรารู้จุดเริ่มต้นของข้อมูลได้จากตัวแปร ShapeHead ตามลำดับที่คำนวณได้และสิ้นสุดเมื่อค่าตัวแปร ShapePoly ที่ตรวจพบมีค่าเป็น 86 (เก็บไว้ตอนสร้างอักขระจากบทความตอนที่ 1) ทำให้เราสามารถนำข้อมูลของแต่ละตัวอักขระนั้น ๆ มาแสดงบนจอภาพได้ถูกต้อง และเนื่องจากเราได้กำหนดให้การเก็บค่าลงข้อมูลเป็นการอ้างอิงกับคู่ลำดับแบบเวกเตอร์ (มีทั้งค่าบวกและลบ) จึงสามารถทำให้เปลี่ยน-แปลงขนาดของอักขระได้โดยที่ยังมีความละเอียดของตัวอักขระเหมือนเดิม พร้อมกับยังสามารถทำให้เป็นอักขระมีรูปร่างต่าง ๆ กันได้ เช่น ตัวขยาย 2 เท่า, ตัวสูง 2 เท่า ฯลฯ

ชุดคำสั่งที่ใช้ในการแสดงอักขระแต่ละตัวอยู่ในชุดคำสั่ง ShowPoly ในรูปที่ 3 ขอให้ทำความเข้าใจด้วยตนเองก่อน

โปรแกรมหลักแสดงอักขระ

เราจำเป็นต้องเขียนชุดคำสั่งเพื่อที่จะได้นำข้อความที่เรากำหนดนำไป

```
PROCEDURE ShowPoly(Xx,Yy:integer; size:byte);
var
    Poly:PolygonType;       P,NO,Patt:byte;
    x,y,Xold,YOld,Ystart,XItalic,Udpix:Integer;

 begin
      udpix:=0;    Ystart:=Yy;
      if up then Udpix:=-size*15;     if down then Udpix:=size*13;
      if TableNO in [156,160] then inc(xx,6*size);
      Xx:=Xx-Size*PolyPoint[1][1];    Yy:=Yy-Size*PolyPoint[1][2];
      if PolyPoint[1][1]=0 then XRight:=Xx;
      Xold:=Xx; YOld:=Yy;  No:=2;     SetPattern;
 Repeat P:=1;
     Repeat
        x:=Xx+Size*PolyPoint[No][1]*Enlarge;
        y:=Yy-Size*PolyPoint[No][2]*high;
        Poly[p].x:=x;     Poly[p].y:=y+udpix;
        if (PolyPoint[1][1]=0) and (XRight<x) and
                      ((YPoint-y)<size*18*high) then  XRight:=x;
        if Italic then  Poly[p].x:=x+round((Ystart-y) div 4);
        case PlayMode of
           1:if x>=0 then Poly[p].y:=Poly[p].y+Ywave[x mod (getMaxX+1)];
           2:if x>=-Size*25 then
                 Poly[p].y:=Poly[p].y-YScope[NoSTR]^[x-Xpoint[NoSTR]+20];
           3:if x>=-Size*25 then
                 Poly[p].y:=Poly[p].y+YScope[NoSTR]^[x-Xpoint[NoSTR]+20];
           4:Poly[p].y:=Poly[p].y+round((x-Xstart)/12);
           5:Poly[p].y:=Poly[p].y-round((x-Xstart)/12);
        end;
        Inc(NO); Inc(p); Xx:=x; Yy:=y;
     until PolyPoint[No][1] in [86,99];
      if xx>=-Size*25 then  FillPoly(P-1, Poly);
      if PolyPoint[No][1]=99 then                  begin
         Xx:=Xold; Yy:=Yold; Inc(No); SetPattern;
         if PolyPoint[No][1]=127 then      begin
          Patt:=PattNo; PattNo:=0; SetPattern;
          PattNo:=Patt; Inc(No);           end;  end;
  Until PolyPoint[No][1]=86;
end;
```

รูปที่ 3 ชุดคำสั่งในการแสดงอักขระตามขนาดที่กำหนด

แสดงข้อความบนจอภาพให้ถูกต้องตรงกับความต้องการของเรา โดยต้องคำนึงถึงสิ่งที่เราได้กำหนดไว้แล้ว เช่น รูปแบบการแสดงข้อความ การเลือกฟอนต์ ตัวหนังสือ ฯลฯ นั้นคือเราต้องกำหนดตัวแปรไว้ใช้งานด้วย

ผู้เขียนได้กำหนดตัวแปรและรหัสในการทำงานไว้ดังนี้

1. ตัวแปร PlayMode=1 และรหัส ^PW (#23) ใช้ในการแสดงข้อความแบบคลื่น

2. ตัวแปร PlayMode=2 และรหัส ^PC (#3) ใช้ในการแสดงข้อความแบบโค้งนูนขึ้น

3. ตัวแปร PlayMode=3 และรหัส ^PY (#25) ใช้ในการแสดงข้อความแบบโค้งเว้าลง

4. ตัวแปร PlayMode=4 และรหัส ^PD (#4) ใช้ในการแสดงข้อความแบบเฉียงลง

5. ตัวแปร PlayMode=5 และรหัส ^PU (#21) ใช้ในการแสดงข้อความแบบเฉียงขึ้น

6. ตัวแปร Enlarge และรหัส ^PE (#5) สำหรับแสดงตัวแบบกว้าง 2 เท่าหรือตัวขยาย

7. ตัวแปร High และรหัส ^PH (#8) สำหรับตัวสูง 2 เท่า

8. ตัวแปร Italic และรหัส ^PL (#12) สำหรับแสดงตัวเอน

9. ตัวแปร Shadow และรหัส ^PS (#19) สำหรับแสดงตัวนูน

10. ตัวแปร Symbol และรหัส ^PP (#16) สำหรับการแสดงอักขระพิเศษ

11. ตัวแปร Up และรหัส ^PT (#20) สำหรับการแสดงแบบตัวยกกำลัง

12. ตัวแปร Down และรหัส ^PV (#22) สำหรับการแสดงแบบตัวห้อยท้าย

13. ตัวแปร TableNo เก็บลำดับของข้อมูล (เปลี่ยนมาจากรหัสแอสกีของอักขระในข้อความอีกที) ซึ่งทำให้เราสามารถกำหนดเงื่อนไขได้ง่ายขึ้น โดยการดูจากตารางในตอนที่ 1 รูปที่ 7

14. ตัวแปร no, noback เก็บลำดับของตัวอักขระที่กำลังตรวจสอบและลำดับที่ได้แสดงไปแล้วก่อนนี้ เพื่อมากำหนดการแสดงอักขระ (บางตัว) ให้มีความเหมาะสมยิ่งขึ้น

15. ตัวแปร font และรหัส ^P\, ^P], ^P^, ^P_ (#28 – #31) เพื่อเลือกฟอนต์ของตัวอักขระ

นำมาเขียนชุดคำสั่งในการทำงานได้ดังรูปที่ 4 ขอให้ทำความเข้าใจในแต่ละบรรทัดด้วยตนเองให้เข้าใจก่อน

```
PROCEDURE ShowBanner(x,y:Integer; Content:string; Size,Patt:byte);
var k,No,B,kk,Exit,TableNoOld,noback,ss:byte;
    Shape:word; xpix,Ypix:integer;

Begin   PattNo:=Patt;     Content:=Content+#255;  Ypoint:=y;
        Shadow:=False;    Italic:=False; up:=false; down:=false;
        Enlarge:=1;       High:=1;  PlayMode:=0;  Symbol:=false;
        if size<3 then ss:=size else ss:=3;
        k:=pos(#3,Content);
        if k<>0 then                              begin
           delete(Content,k,1); PlayMode:=2 end;
        k:=pos(#25,Content);
        if k<>0 then                               begin
           delete(Content,k,1); PlayMode:=3 end;
        if (PlayMode in [2,3]) and (PageNo=1) then  begin
                ScopePoint(content,size);
                xpoint[NoSTR]:=x;                    end;
        k:=pos(#21,Content);
        if k<>0 then                                 begin
             delete(Content,k,1); PlayMode:=5;       end;
        k:=pos(#4,Content);
        if k<>0 then                                 begin
             delete(Content,k,1); PlayMode:=4;  end;
        if (PlayMode in [4,5]) and (PageNo=1) then  Xstart:=x;
        k:=pos(#23,Content);
        if k<>0 then                                begin
             delete(Content,k,1); PlayMode:=1;  end;
        k:=pos(#207,content);
        if k<>0 then begin                                          begin
           if content[k+1] in [#224..#227] then
              content[k]:=content[k+1]; content[k+1]:=#206; inc(k); end;
           content[k]:=#206; insert(#222,content,k);
                end;
        k:=1;  Exit:=0;  setlinestyle(0,123,3);                          ➤
        if shadow or (Size<4) then setlinestyle(0,123,1);
```

รูปที่ 4 ชุดคำสั่งหลักในการนำข้อความไปแสดงในรูปแบบที่กำหนด

```
Repeat
  repeat
    if Content[k] in [#28..#31] then     begin
          font:=ord(Content[k])-27;
          if font>MaxFont then font:=1;  end;
    if Content[k]=#5  then                                  begin
          inc(Enlarge); if Enlarge=3 then Enlarge:=1;  end;
    if Content[k]=#12 then  if Italic then Italic:=false
                              Else Italic:=true;
    if Content[k]=#16 then  if Symbol then Symbol:=false
                              Else Symbol:=true;
    if Content[k]=#19 then  if Shadow then Shadow:=false
                              Else Shadow:=true;
    if Content[k]=#20 then  if Up then Up:=false
                              Else Up:=true;
    if Content[k]=#22 then  if Down then Down:=false
                              Else Down:=true;
    if Content[k]=#8 then                    begin
        inc(High); if High=3 then High:=1; end;
    if Content[k]<#32 then inc(k);
  until Content[k] in [#32..#255];
    if Content[k] in [#32,#160] then Inc(x,20*Size)  else
     begin
          no:=Ord(Content[k])-32;  noback:=Ord(Content[k-1])-32;
      if Content[k]>#160 then            begin
         no:=Ord(Content[k])-65;
         noback:=Ord(Content[k-1])-65;  end;
      if Symbol and (content[k] in [#33..#123]) then inc(no,163);
        Shape:=ShapeHead[font]^[No]-1;  xpix:=0;  Ypix:=0; TableNo:=no;
      if TableNo in [145..147] then Dec(x,8*Size);
      if TableNo in [159..163] then         begin
         if noback= 142 then xpix:=-Size*2;
         if (noback in [142,152..157,222]) or (content[k+1]=#222) then begin
          inc(Ypix,size*8*high);
          if Italic then inc(x,size*2);     end;                              end;
     if (TableNo in [100,101,109,116,119,120,123,124,128,131,133,135,136])
          and (Font=1) then
              if noback in [133,143,144,145..147] then dec(x,size*4);
         B:=0;
           Repeat   Inc(B);  Inc(Shape);
             PolyPoint[B][1]:=ShapePoly[font]^[Shape][1];
             PolyPoint[B][2]:=ShapePoly[font]^[Shape][2];
           until ShapePoly[font]^[Shape][1]=86;
         if shadow then
             for kk:=ss downto 1 do ShowPoly(x+xpix+kk,y-Ypix+kk,size);
             ShowPoly(x+xpix,y-Ypix,size);
           if PolyPoint[1][1]=0 then                    begin
              TableNoOld:=TableNo; x:=XRight+Size*5;  end;
      end;
          inc(k);
       if X>GetMaxX then
          begin
            Position:=k-1;
            if Content[Position]>#160 then no:=Ord(Content[Position])-65
                                      Else no:=Ord(Content[Position])-32;
            if Symbol and (content[Position] in [#33..#123]) then inc(no,131)
            if ShapePoly[font]^[ShapeHead[font]^[No]][1]=0 then Inc(Exit);
            if Exit=2 then k:=length(Content);
          end;
      Until k>length(Content)-1;
        if Exit=0 then Position:=length(Content)+1;                          ■
   end;
```

รูปที่ 4 (ต่อ)

การทำงานของชุดคำสั่ง ShowBanner

1. ส่งค่า x,y ข้อความ, ขนาดของข้อความและรูปแบบพื้นไปใช้ในชุดคำสั่ง

2. ตรวจสอบรหัสรูปแบบการแสดงผล (รหัส ^PC, ^PY, ^PW, ^PD, ^PU) เพื่อกำหนดค่า PlayMode

3. ตรวจการเขียน ำ ้ า ็ ่ เพื่อจะสลับที่ให้ตรงกับการที่จะนำข้อความไปแสดง

4. ตรวจการกำหนดลำดับของฟอนต์ที่จะแสดง (ถ้าไม่กำหนดก็ใช้ฟอนต์เดิมซึ่งกำหนดไว้แล้ว)

5. ตรวจลักษณะการแสดงอักขระ (รหัส ^PL, ^PP, ^PS, ^PT, ^PV)

6. ถ้าตรวจพบว่าเป็นอักขระแอสกี 32,160 ก็ให้ขยับค่า x ไปเท่ากับ 20xsize

7. ถ้าเป็นรหัสอื่น ๆ ก็ให้แสดง

อักขระนั้น ๆ ตามลำดับการทำงานดังนี้

7.1 ตรวจสอบหาลำดับที่ของอักขระเพื่อหาจุดเริ่มต้นของข้อมูลจาก ShapeHead[font]^[no]

7.2 มีการเปลี่ยนแปลงค่า x ในบางอักขระเพื่อให้ได้การแสดงผลมีความสวยงาม ส่วนนี้อยู่ที่จะกำหนดเอาเอง (ไม่จำเป็นต้องตรงกับที่ผู้เขียนกำหนดไว้ก็ได้)

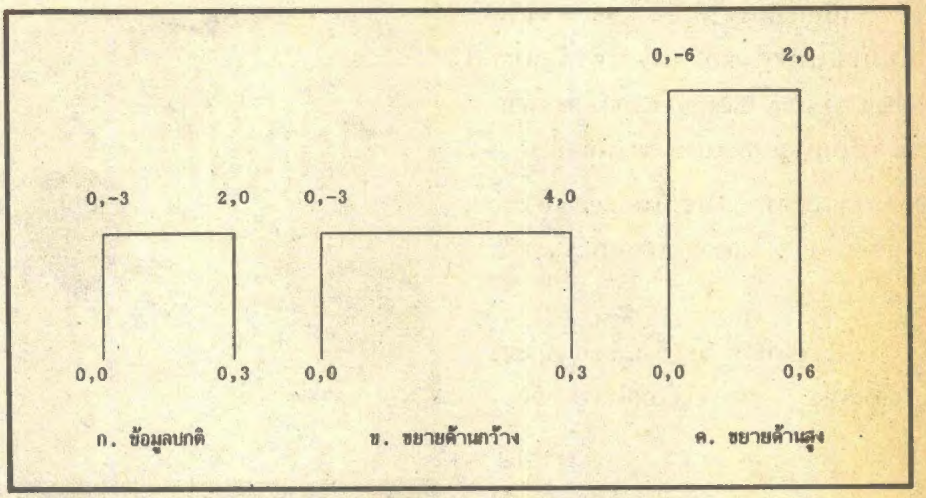
7.3 เริ่มนำข้อมูลจากจุดเริ่มต้นมาใส่ในตัวแปร PolyPoint ทีละหนึ่งจุดจนกว่าจะพบค่า 86 ก็จะเรียกชุดคำสั่ง showPoly เพื่อนำอักขระนั้น ๆ ไปแสดงบนจอภาพ หากมีการกำหนดเป็นตัวนูนก็ให้แสดงตัวนั้นหลาย ๆ ครั้ง ณ ตำแหน่งที่เหมาะสม

7.4 ถ้าตรวจพบว่าข้อมูลลำดับแรก (ตัวแปร PolyPoint[1] [1]) มีค่าเท่ากับ 0 ให้เพิ่มค่า x ขึ้นอีก Sizex5 ซึ่งจะเป็นช่องว่างระหว่างตัวอักขระ ข้อมูลนี้เรากำหนดไว้ตั้งแต่สร้างอักขระในตอนที่ 1 ซึ่งถ้าเราเลือกตำแหน่งเริ่มต้นเป็น 2 หรือ 3 ก็ไม่มีการเพิ่มช่องว่างนี้

7.5 เลื่อนไปตัวอักขระที่อยู่ถัดมาจนกว่าจะหมดข้อความ แต่ถ้าค่า x มากกว่าขอบเขตของจอภาพก็จะสิ้นสุดการทำงานของชุดคำสั่งนี้

การแสดงข้อความแบบขยายและตัวสูง 2 เท่า

เพื่อความเข้าใจในเรื่องการแสดงตัวอักขระแต่ละรูปแบบนั้น ผู้เขียนขอสมมุติข้อมูลของอักขระแบบง่าย ๆ เพียง 4 คู่ลำดับเพื่อประกอบการอธิบายดังรูปที่ 5 โดยคำอธิบายต่อไปนี้อาจมีความเกี่ยวพันกับโปรแกรมในรูปที่ 3



รูปที่ 5 แสดงข้อมูลตามปกติ ข้อมูลขยายด้านกว้าง และข้อมูลขยายด้านสูง

และ 4 ด้วย จึงขอให้ตรวจสอบจุดที่ได้อธิบายของแต่ละคำสั่งเอาเอง ซึ่งตัวเลขที่เก็บไว้ในไฟล์ข้อมูลจะเป็นดังชุด ก. ซึ่งจะได้ข้อมูลดังนี้

จุดที่ 1 เก็บค่า 0,0 ซึ่งถือว่าเป็นจุดเริ่มต้น

จุดที่ 2 เก็บค่า 0,-3 (ค่าที่เก็บนี้เป็นการเก็บเทียบกับจุดที่ 1) หมายถึงว่าค่าในแกน x ไม่เปลี่ยนแปลง แต่ค่าในแกน y จะขยับขึ้นไปข้างบน 3 จุด

จุดที่ 3 เก็บค่า 2,0 (ค่าที่เก็บนี้เป็นการเก็บเทียบกับจุดที่ 2) นั่นคือจุดนี้อยู่ระดับเดียวกันกับจุดที่ 2 แต่จะห่างไปทางขวา 2 จุด

จุดที่ 4 เก็บค่า 0,3 (ค่าที่เก็บนี้เป็นการเก็บเทียบกับจุดที่ 3) ซึ่งแสดงว่าจุดที่ 4 อยู่ด้านล่างของจุดที่ 3 ห่างลงมา 3 จุด

เมื่อจบข้อมูลนี้จะต้องมีการขยับจุดเริ่มต้นไปทางขวาด้วยเพราะข้อมูลของจุดที่ 1 มีค่าเป็น 0,0 ถ้านำข้อมูลของอักขระนี้มาแสดงบนจอภาพจะแสดง ณ ตำแหน่งที่คำนวณได้ (สมมุติตอนนั้นคู่ลำดับของจุดเริ่มต้นเป็น 200, 100) ถ้าแสดงตามปกติค่าตัวแปร High=1 และ Enlarge=1 โดยที่ตัวแปร High จะกระทำกับข้อมูลในแกน y และตัวแปร Enlarge จะกระทำกับข้อมูลในแกน x

1. จุดที่ 1 จะเป็น 200+0×1, 100+0×1 = 200+0,100+0 = 200, 100

2. จุดที่ 2 จะเป็น 200+0×1, 100−3×1 = 200+0,100−3 = 200, 97

3. จุดที่ 3 จะเป็น 200+2×1, 97+0×1 = 200+2,97+0 = 202,97

4. จุดที่ 4 จะเป็น 202+0×1, 97+3×1 = 202+0,97+3 = 202,100

เมื่อกำหนดให้เป็นตัวขยายจะได้ข้อมูลแบบชุด ข. เพราะค่าตัวแปร High=1 และ Enlarge=2 จะได้จุดตามลำดับในการแสดงบนจอภาพดังนี้

1. จุดที่ 1 จะเป็น 200+0×2, 100+0×1 = 200+0,100+0 = 200, 100

2. จุดที่ 2 จะเป็น 200+0×2, 100−3×1 = 200+0,100−3 = 200, 97

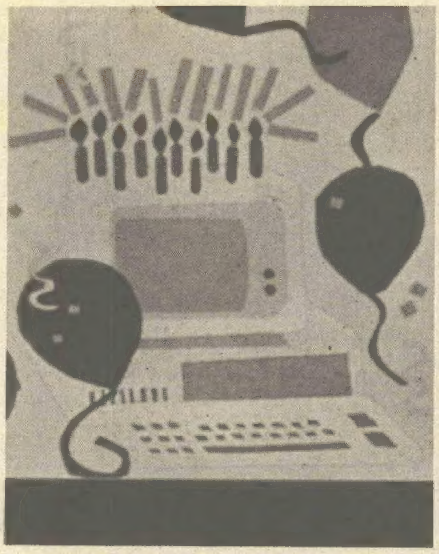
3. จุดที่ 3 จะเป็น 200+2×2, 97+0×1 = 200+4,97+0 = 204,97

4. จุดที่ 4 จะเป็น 204+0×2, 97+3×1 = 204+0,97+3 = 204,100

เมื่อกำหนดให้เป็นตัวสูง 2 เท่า ได้ข้อมูลแบบชุด ค. เพราะค่าตัวแปร High=2 และ Enlarge=1 จะได้จุดตามลำดับในการแสดงบนจอภาพมีค่าดังนี้

1. จุดที่ 1 จะเป็น 200+0×1, 100+0×2 = 200+0,100+0 = 200, 100

2. จุดที่ 2 จะเป็น 200+0×1, 100−3×2 = 200+0,100−6 = 200, 94



3. จุดที่ 3 จะเป็น 200+2×1, 94+0×2 = 200+2,94+0 = 202,94

4. จุดที่ 4 จะเป็น 202+0×1, 94+3×2 = 202+0,94+6 = 202,100

ในการทำตัวเอนนั้นใช้วิธีกำหนดคู่ลำดับในแกน x มีค่าเปลี่ยนไปดังนี้

1. กำหนดจุดเริ่มต้นตามแกน y (ตามตัวอย่างคือ y=100)

2. ถ้าจุดในแกน y มีค่าไม่เท่ากับจุดเริ่มต้น ก็จะมีการเปลี่ยนแปลงค่าตามแกน x คือ

2.1 ถ้าน้อยกว่าจุดเริ่มต้นก็เพิ่มค่าในแกน x

2.2 ถ้ามากกว่าจุดเริ่มต้นก็ลดค่าในแกน x

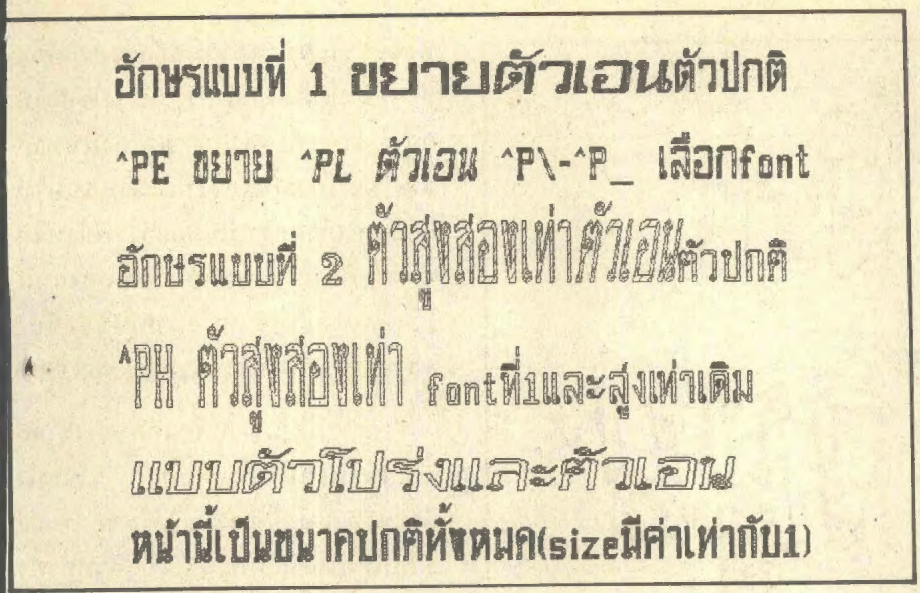
ค่าที่เปลี่ยนแปลงนี้ตามโปรแกรมในรูปที่ 3 จะกำหนดค่าที่เลื่อนจาก (Ystart-y) div 4 ถ้าต้องการให้เอียงมากหรือน้อยกว่านี้ก็ให้เปลี่ยนแปลงค่าเลข 4 ที่เอาไปหาร ผลคือถ้าค่า y น้อยกว่า ystart จะได้ค่าที่คำนวณนี้เป็น (+) จึงทำให้ได้ค่าตามแกน x เพิ่มขึ้น ถ้าค่า y มากกว่า ystart ก็จะได้ค่าเป็น (−) ค่าตามแกน x จึงลดลง (กรณีของการแสดงสระ ุ ู หรืออักขระ ญ ฐ ฏ ฏ เป็นต้น)

ส่วนตัวแบบโปร่งทำได้ง่ายด้วยการกำหนดรูปแบบพื้นเป็น 0 เท่านั้น ตัวอย่างการแสดงข้อความแบบที่ 1 จึงเป็นดังรูปที่ 6 และได้ผลที่ลอกจากจอภาพดังรูปที่ 7 ขอให้สังเกตรหัสที่ใช้ควบคุมการแสดงผลในโปรแกรมด้วย

```
PROCEDURE ShowContent1;
var x,y:integer; size,Pattern:byte;
    ShapeSTR:string;
begin
   size:=1; clsGR(0,0,GetMaxX,GetMaxY);
   pattern:=1;   x:=0;  y:=50;
   ShapeSTR:='อักษรแบบที่ 1 '#5'ขยาย'#12'ตัวเอน'#5#12'ตัวปกติ';
   showBanner(x,y,ShapeSTR,size,pattern);
   pattern:=10;   x:=0;  y:=100;
   ShapeSTR:='^PE ขยาย'#12' ^PL ตัวเอน'#12' ^P\-^P_ เลือกfont';
   showBanner(x,y,ShapeSTR,size,pattern);
   pattern:=13;   x:=0;  y:=160; size:=1;
   ShapeSTR:=#29'อักษรแบบที่ 2 '#8'ตัวสองเท่า'#12'ตัวเอน'#8#12'ตัวปกติ';
   showBanner(x,y,ShapeSTR,size,pattern);
   pattern:=20;   x:=0;  y:=230;
   ShapeSTR:=#8'^PH ตัวสองเท่า'#8#28' fontที่1และสองเท่าเดิม';
   showBanner(x,y,ShapeSTR,size,pattern);
   pattern:=0;   x:=10;  y:=280;
   ShapeSTR:=#12#5'แบบตัวโปร่ง'#29'และตัวเอน';
   showBanner(x,y,ShapeSTR,size,pattern);
   pattern:=5;   x:=10;  y:=320;
   ShapeSTR:='หน้านี้เป็นขนาดปกติที่ยังหมด(sizeมีค่าเท่ากับ1)';
   showBanner(x,y,ShapeSTR,size,pattern);
   readchr;
end;
```

รูปที่ 6 ตัวอย่างการใช้รหัสควบคุมในการแสดงข้อความแบบต่างๆ

อักษรแบบที่ 1 ขยายตัวเอนตัวปกติ
^PE ขยาย ^PL ตัวเอน ^P\-^P_ เลือกfont
อักษรแบบที่ 2 ตัวสูงสองเท่าตัวเอนตัวปกติ
^PH ตัวสูงสองเท่า fontที่1และสูงเท่าเดิม
แบบตัวโปร่งและตัวเอน
หน้านี้เป็นขนาดปกติทั้งหมด(sizeมีค่าเท่ากับ1)



รูปที่ 7 ภาพที่ได้จากหน้าจอกราฟิกของชุดคำสั่งในรูปที่ 6

```
PROCEDURE ShowContent2;
var x,y:integer; size,Pattern:byte;
    ShapeSTR:string;
begin
    clsGR(0,0,GetMaxX,GetMaxY);
    size:=2;  pattern:=19;  x:=0;  y:=45;
    ShapeSTR:=#29#4#19'ความลาดชันคือ';
    showBanner(x,y,ShapeSTR,size,pattern);
    pattern:=20;  x:=0;      y:=170;
    ShapeSTR:=#28#21#19'แนวทั้งสองตำไม่เท่ากัน';
    showBanner(x,y,ShapeSTR,size,pattern);
    pattern:=0;  x:=235;      y:=298;
    ShapeSTR:=#29#12#19'ตัวนูน';  size:=5;
    showBanner(x,y,ShapeSTR,size,pattern);
    pattern:=18;  x:=0; y:=220;
    ShapeSTR:=#5#19'^PSตัวนูน';  size:=1;
    showBanner(x,y,ShapeSTR,size,pattern);
    pattern:=15;  x:=0; y:=290;
    ShapeSTR:=#21'^PU เฉียงขึ้น';
    showBanner(x,y,ShapeSTR,size,pattern);
    pattern:=16;  x:=0; y:=320;
    ShapeSTR:=#4'^PD '#5'เฉียงลง';
    showBanner(x,y,ShapeSTR,size,pattern);
    readchr;
end;
```

รูปที่ 8 ตัวอย่างการแสดงข้อความแบบเฉียงขึ้น เฉียงลงและตัวนูน

แสดงข้อความแบบเฉียงขึ้นลงและตัวนูน

เนื่องจากการเก็บข้อมูลของแต่ละอักขระนั้นมีรูปแบบดังที่ได้แสดงในรูปที่ 5 ดังนั้นเราจึงสามารถนำมาแสดงบนจอภาพได้หลาย ๆ รูปแบบ เช่น การแสดงตัวอักขระแบบเฉียงขึ้นและแบบเฉียงลง ฯลฯ ทั้งนี้เพราะเราใช้หลักที่ว่าตัวอักขระทุกตัวมีมุมล่างซ้าย ณ ตำแหน่งเดียวกัน (จุดนี้ไม่จำเป็นว่าจะต้องลงจุดเสมอไป) เพียงแต่จะมีจุดเริ่มต้นที่ไม่ใช่จุดเดียวกัน ในที่นี้ผู้เขียนหมายความว่าทุกตัวอักขระจะมีข้อมูลลำดับที่ 1 อ้างอิงจากจุดเดียวกันเพียงแต่จุดเริ่มต้น (คู่ลำดับที่ 2) จะเป็นไปตามข้อมูลตอนสร้างขึ้นมา โดยขอให้สังเกตคำสั่งในรูปที่ 3 ในชุดคำสั่ง ShowPoly ที่หาค่าเริ่มต้นดังนี้

```
Xx: = Xx-Size*PolyPoint[1][1];
Yy = Yy-Size*PolyPoint[1][2];
```

แสดงข้อความตามแนวเฉียง

1. กำหนดจุดเริ่มต้นของข้อความตามแนวราบ (แกน x) ในตัวแปรชื่อ Xstart เพื่อนำไปกำหนดการแสดงในแนวเฉียงขึ้นหรือลง

2. เมื่อตรวจพบรหัส ^PD (#4) จะทำให้ตัวแปร PlayMode=4 หรือพบ ^PU (#21) ก็จะได้ตัวแปร PlayMode=5 (กำหนดพร้อมกันสองรหัสจะได้ตัวแปร PlayMode=4 เพราะตรวจรหัส ^PD ทีหลัง)

3. เมื่อนำอักขระนี้ไปแสดงผลด้วยชุดคำสั่ง ShowPoly จะทำการบวกค่าของคู่ลำดับในแกน y ให้เป็นค่าใหม่ (ค่าในแกน x คงเดิม) ตามคำสั่ง

Poly[p].y: = Poly[p].y + round (x-Xstart)/12); ในกรณีที่แสดงในแนวเฉียงลง

Poly[p].y: = Poly[p].y-round (x-Xstart)/12); ในกรณีที่แสดงในแนวเฉียงขึ้น

หากต้องการให้แนวการเฉียงขึ้นหรือลงมีความเอียงผิดจากนี้ให้เปลี่ยนตัวเลข 12 ไปตามความต้องการได้ นั่นคือ อาจกำหนดให้แสดงเฉียงได้หลาย ๆ แนว ส่วนการแสดงตัวนูนเราใช้วิธีง่าย ๆ คือให้แสดงตัวอักขระนั้นซ้ำหลาย ๆ ครั้งในตำแหน่งที่เหมาะสม โปรแกรมจะใช้วิธีแสดงตามคู่ลำดับที่ขยับทั้งแกน x และ y ครั้งละ 1 จุด ซึ่งก็ทำให้ได้ข้อความแบบตัวนูนได้

การกำหนดตัวแปร size ก็จะทำให้ได้ขนาดของอักขระที่มีขนาดต่าง ๆ กันออกไป ตัวอย่างการแสดงผลทั้งสามแบบนี้เป็นดังรูปที่ 8 และได้ภาพที่ลอกจากหน้าจอดังรูปที่ 9

รูปที่ 9 ภาพที่ลอกจากหน้าจอกราฟิกตามโปรแกรมในรูปที่ 8

การแสดงข้อความแบบคลื่นและแนวโค้ง

ในแนวทางเดียวกันเราสามารถเอาข้อความมาแสดงในแนวโค้งได้ ในตัวอย่างที่ให้มานี้จะมีอยู่เพียง 3 แบบ คือ แบบโค้งนูนขึ้น โค้งเว้าลงและแสดงแบบคลื่น โดยอาศัยหลักการดังนี้

1. ตรวจสอบรหัสการแสดงผล คือ ^PC (#3)-โค้งนูน, ^PY (#25) -โค้งเว้า, ^PW (#23)-แบบคลื่น

2. คำนวณหาแนวของคลื่นไว้ล่วงหน้า ตามที่อธิบายในรูปที่ 10 ชุดคำสั่ง WavePoint ส่วนในการแสดงแบบโค้งนั้นจะต้องคำนวณตามข้อความที่แสดง เพราะมีความประสงค์จะให้ได้ความโค้งที่พอดีกันทั้งเริ่มต้นและจบข้อความ จึงต้องส่งข้อความนั้น ๆ ไปคำนวณหาค่า YScope ตามชุดคำสั่ง ScopePoint ในรูปที่ 10 เช่นเดียวกัน

สำหรับค่าของ YScope ตามที่แสดงตัวอย่างนี้อาจจะยังไม่ถูกต้องนัก ทั้งนี้เพราะขนาดความกว้างของแต่ละอักขระไม่เท่ากันแต่การคำนวณใช้ความกว้าง 20*size เท่ากันหมด ดังนั้นหากข้อความมีความโค้งไม่เหมาะสมก็ให้แก้ปัญหาด้วยการเพิ่มช่องว่างหน้าหรือหลังข้อความเอาเอง กรณีที่แสดงแบบโค้งต้องจดจำตำแหน่งเริ่มต้นของข้อความไว้ในตัวแปร Xpoint[NoSTR];

3. เมื่อมีการแสดงอักขระเราจะเปลี่ยนแปลงข้อมูลในแกน y โดยนำค่าของตัวแปรที่คำนวณได้ในข้อ 1 มารวมกับค่าของแกน y เดิม ตามคำสั่ง

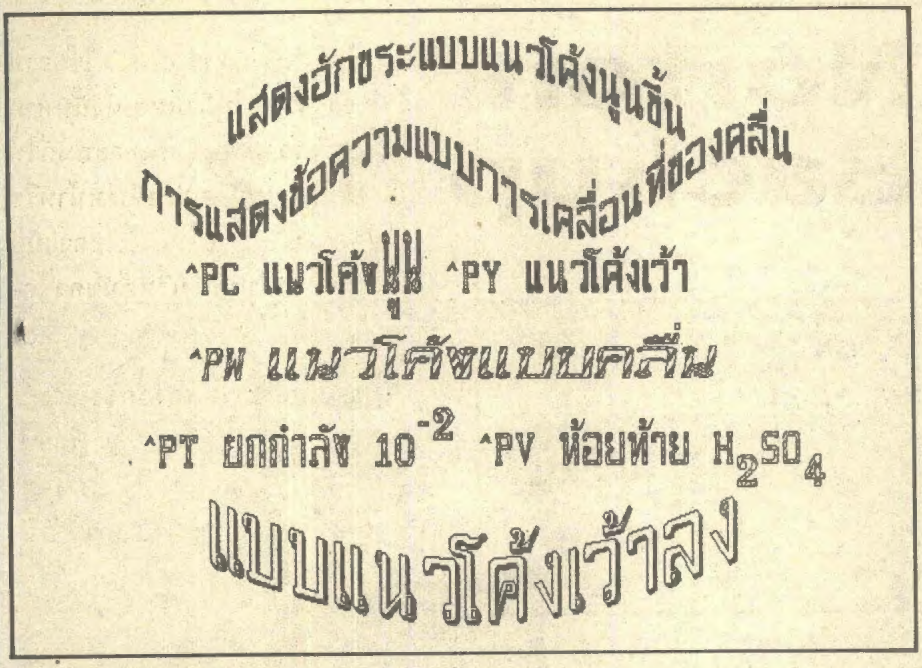
```
Poly[p].y:=Poly[p].y+YWave[x mod (getmaxx+1)];  -แบบคลื่น
Poly[p].y:=Poly[p].y-YScope[NoSTR]^[x-Xpoint[NoSTR]+20];
-แบบโค้งนูน
Poly[p].y:=Poly[p].y+YScope[NoSTR]^[x-Xpoint[NoSTR]+20];
-แบบโค้งเว้า
```

ซึ่งจะทำให้ได้ข้อความที่แสดงตามแบบที่ต้องการได้

ในแต่ละกรณีนี้เรายังสามารถกำหนดให้เป็นตัวเอน ตัวนูน ตัวขยาย หรือตัวสูงได้อีกด้วย จึงทำให้รูปแบบของอักขระที่เก็บไว้เพียงแบบเดียวนี้

```
PROCEDURE ShowContent3;
var ShapeSTR:string;
begin
 clsGR(0,0,GetMaxX,GetMaxY);
 ShapeSTR:=#3'แสดงอักขระแบบแนวโค้งนูนขึ้น ';
 showBanner(80,70,ShapeSTR,1,11);
 ShapeSTR:=#23#28'การแสดงข้อความแบบการเคลื่อนที่ของคลื่น';
 showBanner(0,95,ShapeSTR,1,6);
 ShapeSTR:=#29'^PC แนวโค้ง'#8'นูน'#8#28' ^PY แนวโค้งเว้า';
 showBanner(40,150,ShapeSTR,1,7);
 ShapeSTR:=#29#12'^PW '#5'แนวโค้งแบบคลื่น';
 showBanner(40,200,ShapeSTR,1,14);
 ShapeSTR:='^PT ยกกำลัง 10'#20'-2'#20#28' ^PV หดยาย H'#22'2'#22'SO'#22'4';
 showBanner(0,250,ShapeSTR,1,18);
 ShapeSTR:=#25#19' แบบแนวโค้งเว้าลง ';
 showBanner(20,290,ShapeSTR,2,0);
 readchr;
end;
```

รูปที่ 10 ตัวอย่างการแสดงข้อความแบบโค้งนูน โค้งเว้าและแบบคลื่น



รูปที่ 11 ภาพที่ได้จากหน้าจอกราฟิกของชุดคำสั่งในรูปที่ 10

```
{======================= THAI-BANNER(2) =======================}

begin
opengraph;
   Colorbg:=14; ColorBox:=14;   wavepoint;
   if Monoscreen then begin Colorbg:=1; ColorBox:=1; end;
   for font:=1 to 10 do
     begin
       Mark(Yscope[font]); new(Yscope[font]);
     end;
       LoadFile;   PageNo:=1; NoSTR:=1;
       showContent1;
       showContent2;
       showContent3;
     for font:=1 to MaxFont do
       begin
         release(ShapePoly[font]);
         release(ShapeHead[font]);
       end;
      for font:=1 to 10 do  release(Yscope[font]);
    closegraph;
end.
```

รูปที่ 12 แสดงโปรแกรมหลักเรียกใช้โปรแกรมอื่น ๆ มาทำงาน

แสดงผลบนจอภาพได้มากแบบขึ้น ตัวอย่างในรูปที่ 10 เป็นการแสดงข้อความแบบโค้งนูน โค้งเว้า และแบบคลื่น โดยได้ลอกภาพหน้าจอมาให้ดูเป็นตัวอย่างตามรูปที่ 11 อย่าลืมว่า.....ยังสามารถนำไปแสดงผลแบบอื่น ๆ ได้อีกมากมาย จึงฝากไว้ให้ลองคิดพัฒนากันเอาเอง

ฝากไว้ให้คิด

ตามที่ได้แสดงโปรแกรมมานี้ ผู้อ่านจะเห็นว่าการเลือกใช้โปรแกรมทั้งหมดจะเป็นไปดังรูปที่ 12 ซึ่งต้องกำหนดหน่วยความจำฮีพให้แก่ตัวแปรต่าง ๆ ก่อนจะรันโปรแกรม และแสดงข้อความตามตัวอย่างก่อนจบโปรแกรมก็ต้องลบหน่วยความจำฮีพด้วย

โปรแกรมทั้งหมดนี้ในซอฟต์เมล์บรรจุอยู่ในไฟล์ banner2.pas และมีฟอนต์มาให้ด้วย 2 ฟอนต์และมีปัญหาที่ขอทิ้งไว้ให้ผู้อ่านลองคิดและเขียนโปรแกรมล่วงหน้าไปก่อนที่จะได้อ่านตอนที่ 3 ดังนี้

1. เขียนโปรแกรมเพื่อใช้ในการพิมพ์ภาพอย่างไร จึงจะเป็น "ป้ายประกาศ" ตามต้องการ

2. เมื่อเป็นป้ายประกาศก็ต้องมีความยาวหลายหน้า ปัญหาคือเราจะพิมพ์ข้อความยาว ๆ ของเราลงในกระดาษได้อย่างไร ในเมื่อขนาดของจอภาพในการแสดงข้อความมีจำกัดเพียงขอบเขตของกราฟิกเท่านั้น

3. ตำแหน่งเริ่มต้นของข้อความแต่ละข้อความ (ที่อาจมีหลายหน้า) จะทำอย่างไร

หวังว่ากว่าจะถึงตอนที่สาม ผู้อ่านคงได้แนวคิดในการพิมพ์ลงกระดาษเป็นป้ายประกาศ และได้เขียนโปรแกรมกันไปบ้าง ก่อนจบผู้เขียนขอร้องมายังผู้อ่านสักหนึ่งเรื่องคือ หากได้สร้างอักขระที่มีรูปแบบของอักขระต่างจากที่ผู้เขียนได้ให้มากับซอฟต์เมล์ฉบับนี้ ประสงค์จะเผยแพร่ให้เป็นประโยชน์ต่อผู้อื่นก็ขอช่วยกรุณาก๊อบปี้ไฟล์ส่งไปให้ผู้เขียนด้วยเพื่อผู้เขียนจะได้นำมาใส่ในซอฟต์เมล์ของตอนที่ 3 ต่อไป โดยผู้เขียนขอตอบแทนโดยการก๊อบปี้ซอฟต์เมล์ของตอนที่ 1 และ 2 กลับคืนมาให้ พบกับตอนจบในฉบับหน้า ตอน "การพิมพ์ป้ายประกาศ" ครับ